



PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The version of the following full text has not yet been defined or was untraceable and may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/35849>

Please be advised that this information was generated on 2017-12-06 and may be subject to change.

Freyd is Kleisli, for Arrows

Bart Jacobs^{1,2} Ichiro Hasuo²

Institute for Computing and Information Sciences, Radboud University Nijmegen

Postbus 9010, 6500GL Nijmegen, the Netherlands

<http://www.cs.ru.nl/{B.Jacobs,I.Hasuo}> {b.jacobs, ichiro}@cs.ru.nl

Abstract

Arrows have been introduced in functional programming as generalisations of monads. They also generalise comonads. Fundamental structures associated with (co)monads are Kleisli categories and categories of (Eilenberg-Moore) algebras. Hence it makes sense to ask if there are analogous structures for Arrows. In this short note we shall take first steps in this direction, and identify for instance the Freyd category that is commonly associated with an Arrow as a Kleisli category.

Keywords: Arrow, Freyd category, Kleisli category, Algebra

1. INTRODUCTION

Arrows have been introduced by Hughes [4] as generalisation of monads, in order to capture a common interface for non-trivial programs such as parsers. See also [7, 5, 13, 1] for more applications.

From a categorical point of view, Arrows have long been considered as an alternative formulation of the (older) concept of Freyd category, from [10]. Recently, an alternative categorical description of Arrows has emerged [3], as a monoid in a category of bifunctors $\mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$. This description is closer to the functional one, and enables a precise formulation of the one-to-one relationship between Arrows and Freyd categories, see [3] for details.

This setting of Arrows-as-monoids forms the starting point for the investigations in this paper. It extends the familiar description of monads-as-monoids (see [6, VII.3]) and comonads-as-monoids. Since Kleisli categories and categories of (Eilenberg-Moore) algebras are fundamental constructions for monads and comonads, we ask ourselves the obvious question: are there analogous constructions for Arrows?

In this note we shall give first answers, which are formulated at a high level of abstraction, touching upon a 2-categorical description (like in [11, 9]). Concrete examples are missing at this stage—apart from examples that come from (co)monads. What we do is formulate notions of algebra and Kleisli category for an Arrow, and illustrate that they are reasonable.

In particular, the one-to-one correspondence from Arrows to Freyd categories [3] is identified as the Kleisli construction for Arrows; hence the title “Freyd is Kleisli”. For this notion of Kleisli construction we have two justifications: 1) for Arrows induced by (co)monads, it coincides with the (usual) Kleisli construction for (co)monads; 2) it is characterised as a suitable left 2-adjoint, which is also the case for (co)monads [11]. In the course we extend the notion of Arrows to be defined on Freyd categories, which leads to some additional 2-categorical accounts.

It is straightforward to extend all the notions and results to enriched settings. That is, an Arrow is now a bifunctor $\mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbb{V}$ (instead of $\mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$) where \mathbb{V} is a symmetric monoidal closed category and \mathbb{C} is \mathbb{V} -enriched. It subsumes for example an Arrow $\mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbb{C}$ when \mathbb{C} is

¹Also part-time at Technical University Eindhoven, NL.

²Part of this work was done during the authors' stay at Research Center for Verification and Semantics, National Institute of Advanced Industrial Science and Technology, Japan. They are grateful for the hospitality.

Cartesian closed. This extension is not purely of technical interest: most notions of computations are found in enriched settings such as by cpo's and not in **Sets**. However in this paper we stick to the non-enriched setting for the sake of simplicity of presentation.

This paper is not self-contained. We shall assume familiarity with earlier work on Arrows and their categorical semantics. We briefly recall notation and equations for Arrows in Section 2, together with the main results of [3]: “Arrows as monoids” and “Arrows are Freyd categories”. In Section 3 the notion of Arrow-algebras is introduced and justified. Then we shall identify the construction of a Freyd category associated with an Arrow as a Kleisli construction in Section 4. Only at this point we start using a 2-categorical language. We conclude in Section 5.

2. ARROWS

As introduced in [4], an Arrow is in the notation of the programming language Haskell a type class of the form:

```
class Arrow A where
  arr  :: (X  -> Y) -> A X Y
  (>>>) :: A X Y -> A Y Z -> A X Z
  first :: A X Y -> A (X,Z) (Y,Z)
```

It describes an Arrow as a binary operation $(X, Y) \mapsto A X Y$ —which we shall often write with brackets as $A(X, Y)$ —together with three operations `arr`, `>>>` and `first`. These operations should satisfy the following equations.

$$(a \ggg b) \ggg c = a \ggg (b \ggg c), \quad (1)$$

$$\text{arr}(g \circ f) = \text{arr}(f) \ggg \text{arr}(g), \quad (2)$$

$$\text{arr id} \ggg a = a = a \ggg \text{arr id}, \quad (3)$$

$$\text{first}(a) \ggg \text{arr}(\pi_1) = \text{arr}(\pi_1) \ggg a, \quad (4)$$

$$\text{first}(a) \ggg \text{arr}(\text{id} \times f) = \text{arr}(\text{id} \times f) \ggg \text{first}(a), \quad (5)$$

$$\text{first}(a) \ggg \text{arr}(\alpha) = \text{arr}(\alpha) \ggg \text{first}(\text{first}(a)), \quad (6)$$

$$\text{first}(\text{arr}(f)) = \text{arr}(f \times \text{id}) \quad (7)$$

$$\text{first}(a \ggg b) = \text{first}(a) \ggg \text{first}(b) \quad (8)$$

where we use

$$\pi_i : X_1 \times X_2 \rightarrow X_i, \quad \alpha : X \times (Y \times Z) \xrightarrow{\cong} (X \times Y) \times Z,$$

for the familiar product maps `fst`, `snd` and `assoc`. Also, $\text{arr}(\text{id} \times f)$ is sometimes written as $\text{second}(\text{arr}(f))$, where

$$\text{second}(a) = \text{arr}(\gamma) \ggg \text{first}(a) \ggg \text{arr}(\gamma),$$

and $\gamma : X \times Y \xrightarrow{\cong} Y \times X$ is the well-known `swap`-map.

It is not hard to show that an Arrow is a bifunctor: for maps $f : X' \rightarrow X$ and $g : Y \rightarrow Y'$ one can define:

$$A(X, Y) \xrightarrow{A(f, g)} A(X', Y') \quad \text{by} \quad a \mapsto \text{arr}(f) \ggg a \ggg \text{arr}(g).$$

As shown in [3], Arrows can be understood abstractly as bifunctors $A : \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$, mapping two types X (for input) and Y (for output) to the set/type $A(X, Y)$ of computations taking elements of X to elements of Y . The map `arr` then appears as natural transformation $\text{Hom}_{\mathbb{C}}(-, +) \rightarrow A(-, +)$. Composition `>>>` can also be seen as a natural transformation $A \otimes A \rightarrow A$, for the

standard tensor product \otimes of bifunctors / profunctors / distributors $\mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$, see [2]. In this way one can understand the triple (A, arr, \ggg) as a monoid in a category of bifunctors. Such a monoid thus consists of a functor $A: \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$ and natural transformations

$$A \otimes A \xrightarrow{\ggg} A \xleftarrow{\text{arr}} \text{Hom}$$

satisfying the familiar monoid laws. Naturality of arr and \ggg means that for maps $f: X' \rightarrow X$ and $g: Y \rightarrow Y'$ in \mathbb{C} one has:

$$\begin{aligned} A(f, g)(\text{arr}(h)) &= \text{arr}(g \circ h \circ f) \\ A(f, g)(a \ggg b) &= A(f, \text{id})(a) \ggg A(\text{id}, g)(b). \end{aligned}$$

The latter leads to a useful description of the action $A(f, g)$ as $A(f, g) = \text{arr}(f) \ggg (-) \ggg \text{arr}(g)$, because:

$$\begin{aligned} A(f, g)(a) &= A(f, g)(\text{arr}(\text{id}) \ggg a) \\ &= A(f, \text{id})(\text{arr}(\text{id})) \ggg A(\text{id}, g)(a) \\ &= \text{arr}(f) \ggg A(\text{id}, g)(a \ggg \text{arr}(\text{id})) \\ &= \text{arr}(f) \ggg A(\text{id}, \text{id})(a) \ggg A(\text{id}, g)(\text{arr}(\text{id})) \\ &= \text{arr}(f) \ggg a \ggg \text{arr}(g). \end{aligned}$$

As shown in [3], the map first is equivalent to a certain “internal strength” map $A(X, Y) \rightarrow A(X, X \times Y)$ with suitable (di)naturality requirements. It will play only a minor role in this paper.

A strong monad $(T, \eta, \mu, \text{st})$ on a category \mathbb{C} with finite products induces an Arrow $\overline{T}: \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$ by $(X, Y) \mapsto \text{Hom}_{\mathbb{C}}(X, TY)$. For instance its operator first is given by $\text{first}_Z(a) = \text{st} \circ (a \times Z)$. Similarly, for a comonad (G, ε, δ) we write \overline{G} for the Arrow given by $(X, Y) \mapsto \text{Hom}_{\mathbb{C}}(GX, Y)$. And in case we have both a strong monad T and a comonad G , together with a distributive law $\lambda: GT \Rightarrow TG$ we write the associated Arrow as $\overline{(G, T)}$. It is given by $(X, Y) \mapsto \text{Hom}_{\mathbb{C}}(GX, TY)$.

The following definition gives the obvious way to associate a category of “computations” with an Arrow.

Definition 1 For an Arrow $A: \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbf{Sets}$, with operations arr, \ggg and first , we define an associated category \mathbb{C}_A with objects $X \in \mathbb{C}$ and morphisms $X \rightarrow Y$ given by elements of the set $A(X, Y)$. Identities and composition are given by arr and \ggg in the obvious manner.

We then have a functor $J_A: \mathbb{C} \rightarrow \mathbb{C}_A$ given by arr . This forms an instance of a Freyd category [8]. For example, the pre-monoidal structure \boxtimes of \mathbb{C}_A comes from the operation first ; the functor J_A preserves pre-monoidal structures on-the-nose.

This mapping $A \mapsto (\mathbb{C} \xrightarrow{J_A} \mathbb{C}_A)$ from Arrows to Freyd categories we shall call the **Kleisli construction** for Arrows. The justification for this name is one of the main points of this paper. One way of doing this is as follows; the other 2-categorical one is later in Section 4.

For an Arrow induced by a (co)monad, the associated Freyd category \mathbb{C}_A coincides with the (co)Kleisli category for the (co)monad in a usual sense. For the Arrow $\overline{(G, T)}$ induced by a monad T , a comonad G and a distributive law $GT \Rightarrow TG$, the mapping yields what is called the bi-Kleisli category used e.g. in [12].

The Kleisli construction $A \mapsto (\mathbb{C} \xrightarrow{J_A} \mathbb{C}_A)$ turns out to be a bijective map, from Arrows to Freyd categories. This observation is made in [3], turning the oft-heard (informal) statement “Arrows are Freyd categories” [7] into a concrete one.

Theorem 2 (Theorem 5.4, [3]) For a locally small category \mathbb{C} with finite products, there is a one-to-one correspondence between

1. Arrows A on \mathbb{C} ;
2. Locally small Freyd categories $\mathbb{C} \rightarrow \mathbb{D}$.

Proof. The Kleisli construction for Arrows associates a Freyd category to an Arrow. Conversely, given a Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{D}$, we obtain an Arrow A on \mathbb{C} by: $A(X, Y) = \text{Hom}_{\mathbb{D}}(JX, JY)$. \square

3. EILENBERG-MOORE ALGEBRAS FOR ARROWS

In this section we shall define the notion of Eilenberg-Moore algebras for Arrows. Its justification is made in two ways: 1) for Arrows induced by (co)monads it coincides with a variant of the usual notion of (co)algebras; 2) an Arrow-algebra is the same thing as a retraction of a Kleisli category, much like for monads.

Let us start by the situation for monads. We shall understand (Eilenberg-Moore) algebras (and coalgebras) in 2-categorical style as natural transformations. This means that an algebra for a monad (T, η, μ) on a category \mathbb{C} is a map $\varphi: T \Rightarrow \text{id}_{\mathbb{C}}$ satisfying the familiar equations $\varphi \circ \eta = \text{id}$ and $\varphi \circ T\varphi = \varphi \circ \mu$.

Since such a monad T is the same thing as a monoid in the category of endofunctors $\mathbb{C} \rightarrow \mathbb{C}$ —with monoidal structure given by functor composition and identity functor—we may write these monad equations as diagrams:

$$\begin{array}{ccc} \text{id}_{\mathbb{C}} & \xrightarrow{\eta} & T \xleftarrow{\mu} T \otimes T = T^2 \\ & \searrow \varphi & \downarrow \varphi \otimes \varphi = \varphi \circ T\varphi \\ & & \text{id}_{\mathbb{C}} = \text{id}_{\mathbb{C}} \otimes \text{id}_{\mathbb{C}} \end{array}$$

When T is strong we might as well require a coherence condition with its strength. Namely, for each $X, Y \in \mathbb{C}$,

$$\begin{array}{ccc} TX \times Y & \xrightarrow{\text{st}} & T(X \times Y) \\ & \searrow \varphi \times Y & \downarrow \varphi \\ & & X \times Y \end{array} \quad (9)$$

Recalling that an Arrow is also a monoid—now in the category of bifunctors—we are led to the following notion.

Definition 3 Let $A : \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \text{Sets}$ be an Arrow. An **algebra** for this Arrow is a natural transformation $\chi : A \Rightarrow \text{Hom}$ such that:

- It is compatible with arr and \gg . That is, the following diagrams commute.

$$\begin{array}{ccc} \text{Hom} & \xrightarrow{\text{arr}} & A \xleftarrow{\gg} A \otimes A \\ & \searrow \chi & \downarrow \chi \otimes \chi \\ & & \text{Hom} \xleftarrow{\cong} \text{Hom} \otimes \text{Hom} \end{array} \quad (10)$$

- It is compatible with first . That is, the following diagram commutes for each $X, Y, Z \in \mathbb{C}$.

$$\begin{array}{ccc} A(X, Y) & \xrightarrow{\text{first}_Z} & A(X \times Z, Y \times Z) \\ \chi \downarrow & & \downarrow \chi \\ \text{Hom}(X, Y) & \xrightarrow{- \times Z} & \text{Hom}(X \times Z, Y \times Z) \end{array}$$

An algebra of an Arrow is thus a (natural) mapping of computations to (pure) functions. One might expect to have a single mapping $\chi : A(X, Y) \rightarrow \text{Hom}(X, Y)$, but then it is not clear how to capture commutation with composition \gg . Specifically, it is not clear how we define $\chi \otimes \chi$ in (10).

In elementary terms this means that an algebra χ satisfies the following three equations.

$$\chi(\text{arr}(f)) = f \quad \chi(a \ggg b) = \chi(b) \circ \chi(a) \quad \chi(\text{first}(a)) = \chi(a) \times \text{id}.$$

Naturality is a consequence since:

$$\begin{aligned} \chi(A(f, g)(a)) &= \chi(\text{arr}(f) \ggg a \ggg \text{arr}(g)) \\ &= \chi(\text{arr}(g)) \circ \chi(a) \circ \chi(\text{arr}(f)) \\ &= g \circ \chi(a) \circ f \\ &= \text{Hom}(f, g)(\chi(a)). \end{aligned}$$

Here is a first basic result.

Proposition 4 *For a strong monad T there is a bijective correspondence between strong-monad-algebras $T \Rightarrow \text{id}$ and Arrow-algebras $\bar{T} \Rightarrow \text{Hom}$ for the corresponding Arrow \bar{T} .*

Later on in this section we shall see two similar results. For this first one we write out the proof in full detail, in order to show which parts are trivial and which are not. Especially the long calculation below is a bit tricky in the order in which the equations have to be applied.

Proof. Given a monad algebra $\varphi: T \Rightarrow \text{id}$. We define $\bar{\varphi}: \bar{T} \Rightarrow \text{Hom}$ on $a \in \text{Hom}(X, TY)$ by $\bar{\varphi}_{(X, Y)}(a) = \varphi_Y \circ a$. We check that it satisfies the required properties (omitting subscripts for convenience).

$$\begin{aligned} \bar{\varphi}(\text{arr}(f)) &= \varphi \circ \eta \circ f & \bar{\varphi}(\text{first}_Z(a)) &= \varphi \circ \text{st} \circ (a \times Z) \\ &= f & &= (\varphi \times Z) \circ (a \times Z) \\ \bar{\varphi}(a \ggg b) &= \varphi \circ \mu \circ T(b) \circ a & &= (\varphi \circ a) \times Z \\ &= \varphi \circ T(\varphi \circ b) \circ a & &= \bar{\varphi}(a) \times Z \\ &= (\varphi \circ b) \circ (\varphi \circ a) & \bar{\varphi}(\bar{T}(f, g)(a)) &= \bar{\varphi}(T(g) \circ a \circ f) \\ &= \bar{\varphi}(b) \circ \bar{\varphi}(a) & &= \varphi \circ T(g) \circ a \circ f \\ & & &= g \circ \varphi \circ a \circ f \\ & & &= \text{Hom}(f, g)(\varphi \circ a) \\ & & &= \text{Hom}(f, g)(\bar{\varphi}(a)). \end{aligned}$$

Conversely, given an Arrow-algebra $\chi: \bar{T} \Rightarrow \text{Hom}$ we define $\bar{\chi}_X: TX \rightarrow X$ as $\chi_{(TX, X)}(\text{id}_{TX})$. Then we get a T -algebra. The unit-law and naturality are easy, and done first.

$$\begin{aligned} \bar{\chi} \circ \eta &= \chi(\text{id}) \circ \chi(\text{arr}(\eta)) & \bar{\chi} \circ T(f) &= \chi(\text{id}) \circ \chi(\text{arr}(T(f))) \\ &= \chi(\text{arr}(\eta) \ggg \text{id}) & &= \chi(\text{arr}(T(f)) \ggg \text{id}) \\ &= \chi(\mu \circ T(\text{id}) \circ \eta \circ \eta) & &= \chi(\mu \circ T(\text{id}) \circ \eta \circ T(f)) \\ &= \chi(\eta) & &= \chi(T(f)) \\ &= \chi(\text{arr}(\text{id})) & &= \chi(\mu \circ T(\eta \circ f) \circ \text{id}) \\ &= \text{id} & &= \chi(\text{id} \ggg \text{arr}(f)) \\ & & &= \chi(\text{arr}(f)) \circ \chi(\text{id}) \\ & & &= f \circ \bar{\chi}. \end{aligned}$$

Compatibility (9) with strength is proved as follows.

$$\begin{aligned}
 \overline{\chi} \times Y &= \chi(\text{id}) \times Y \\
 &\stackrel{(*)}{=} \chi(\text{st} \circ (\text{id} \times \text{id})) \\
 &= \chi(\text{st} \circ \text{id}) \\
 &= \chi(\text{id} \circ \text{st}) \\
 &\stackrel{(\dagger)}{=} \chi(\text{id}) \circ \text{st} \\
 &= \overline{\chi} \circ \text{st}.
 \end{aligned}$$

Here the equality $(*)$ is because χ is compatible with first given by $\text{first}(a) = \text{st} \circ (a \times \text{id})$; (\dagger) is because of the naturality of χ .

Verification of the multiplication law is more subtle.

$$\begin{aligned}
 \overline{\chi} \circ T(\overline{\chi}) &= \chi(\text{id}) \circ \chi(\text{arr}(T(\overline{\chi}))) \\
 &= \chi(\text{arr}(T(\overline{\chi})) \ggg \text{id}) \\
 &= \chi(\mu \circ T(\text{id}) \circ \eta \circ T(\overline{\chi})) \\
 &= \chi(T(\overline{\chi})) \\
 &= \chi(\mu \circ T(\eta \circ \overline{\chi}) \circ \text{id}) \\
 &= \chi(\text{arr}(\overline{\chi}) \ggg \text{id}) \\
 &= \chi(\text{id}) \circ \chi(\text{arr}(\overline{\chi})) \\
 &= \chi(\text{id}) \circ \chi(\text{id}) \\
 &= \chi(\text{id} \ggg \text{id}) \\
 &= \chi(\mu \circ T(\text{id}) \circ \text{id}) \\
 &= \chi(\mu) \\
 &= \chi(\mu \circ T(\text{id}) \circ \eta \circ \mu) \\
 &= \chi(\text{arr}(\mu) \ggg \text{id}) \\
 &= \chi(\text{id}) \circ \chi(\text{arr}(\mu)) \\
 &= \overline{\chi} \circ \mu.
 \end{aligned}$$

Finally we check that we really have a bijective correspondence:

$$\begin{array}{ll}
 \overline{\varphi} &= \overline{\varphi}(\text{id}) & \overline{\chi}(a) &= \overline{\chi} \circ a \\
 &= \varphi \circ \text{id} & &= \chi(\text{id}) \circ \chi(\text{arr}(a)) \\
 &= \varphi & &= \chi(\text{arr}(a) \ggg \text{id}) \\
 & & &= \chi(\mu \circ T(\text{id}) \circ \eta \circ a) \\
 & & &= \chi(a).
 \end{array}$$

□

The definition of $\overline{\chi}$ in this proof suggests that the Yoneda Lemma is applicable. We have chosen to write out the proof directly, for clarity.

There is a dual result for comonads. This shows that Arrow-algebras form a common generalisation of monad algebras and comonad coalgebras. The proof is similar to the one above, and left to the reader.

Proposition 5 *For a comonad G there is a bijective correspondence between comonad-coalgebras $\text{id} \Rightarrow G$ and Arrow-algebras $\overline{G} \Rightarrow \text{Hom}$ for the corresponding Arrow \overline{G} .* □

We can even extend such results to bialgebras.

Proposition 6 Assume a strong monad T , a comonad G , and a distributive law $\lambda: GT \Rightarrow TG$ between them. Then there is a bijective correspondence between λ -bialgebras $T \Rightarrow \text{id} \Rightarrow G$ and Arrow-algebras $\overline{(T, G)} \Rightarrow \text{Hom}$.

Proof. We shall only give the essentials, and leave details to the reader. Assume a λ -bialgebra $T \xRightarrow{\varphi} \text{id} \xRightarrow{\psi} G$, where by definition the following diagram commutes.

$$\begin{array}{ccc} T & \xrightarrow{\varphi} & \text{id} \xrightarrow{\psi} G \\ \psi T \downarrow & & \uparrow \varphi G \\ GT & \xrightarrow{\lambda} & TG \end{array}$$

We obtain an Arrow-algebra $\overline{(\varphi, \psi)}: \overline{(T, G)} \Rightarrow \text{Hom}$ by $\overline{(\varphi, \psi)}(a) = \varphi \circ a \circ \psi$. It satisfies the required equations.

In the reverse direction, assume an Arrow-algebra $\chi: \overline{(T, G)} \Rightarrow \text{Hom}$. We define a pair of maps $\bar{\chi} = (\bar{\chi}_1, \bar{\chi}_2)$ as:

$$\begin{aligned} \bar{\chi}_1 &= \chi_{(TX, X)}(\varepsilon_{TX}) & : & TX \longrightarrow X \\ \bar{\chi}_2 &= \chi_{(X, GX)}(\eta_{GX}) & : & X \longrightarrow GX. \end{aligned}$$

and then check that it yields an appropriate λ -bialgebra. Here we only show commutation of the above diagram, *i.e.* that we have a compatible algebra-coalgebra pair.

$$\begin{aligned} \bar{\chi}_1 \circ \lambda \circ \bar{\chi}_2 &= \chi(\varepsilon) \circ \chi(\text{arr}(\lambda)) \circ \chi(\eta) \\ &= \chi(\eta \ggg \text{arr}(\lambda) \ggg \varepsilon) \\ &= \chi([\mu \circ T(\eta \circ \lambda \circ \varepsilon) \circ \lambda \circ G(\eta) \circ \delta] \ggg \varepsilon) \\ &= \chi([\eta \circ \lambda] \ggg \varepsilon) \\ &= \chi(\mu \circ T(\varepsilon) \circ \lambda \circ G(\eta \circ \lambda) \circ \delta) \\ &= \chi(\lambda) \\ &= \chi(\mu \circ T(\eta) \circ \lambda \circ G(\varepsilon) \circ \delta) \\ &= \chi(\varepsilon \ggg \eta) \\ &= \chi(\eta) \circ \chi(\varepsilon) \\ &= \bar{\chi}_2 \circ \bar{\chi}_1. \end{aligned}$$

□

When John Power saw the notion of Arrow-algebra, he immediately conjectured the following “obvious” result.

Lemma 7 An algebra for an Arrow A on \mathbb{C} is the same thing as a left inverse (or retraction) $K: \mathbb{C}_A \rightarrow \mathbb{C}$ of the “Kleisli inclusion” functor $J_A: \mathbb{C} \rightarrow \mathbb{C}_A$ (meaning that $KJ = \text{id}$).

Proof. Given an algebra $\chi: A \Rightarrow \text{Hom}$ we get a functor $\mathbb{C}_A \rightarrow \mathbb{C}$ by $X \mapsto X$ and $a \mapsto \chi(a)$. It forms a retraction because $\chi(\text{arr}(f)) = f$. Conversely, a retraction $K: \mathbb{C}_A \rightarrow \mathbb{C}$ yields an algebra $\bar{K}: A \Rightarrow \text{Hom}$ by $a \mapsto K(a)$. The only thing worth checking is naturality:

$$\begin{aligned} \bar{K}(A(f, g)(a)) &= K(\text{arr}(f) \ggg a \ggg \text{arr}(g)) \\ &= K(\text{arr}(g)) \circ K(a) \circ K(\text{arr}(f)) \\ &= KJ(g) \circ K(a) \circ KJ(f) \\ &= g \circ K(a) \circ f \\ &= \text{Hom}(f, g)(\bar{K}(a)). \end{aligned}$$

□

We claim that the previous lemma justifies our notion of Eilenberg-Moore algebras for Arrows, because for monads we have an analogous characterisation.

For a monad T on a category \mathbb{C} , there is a bijective correspondence between algebras $T \Rightarrow \text{id}$ and retractions of the Kleisli inclusion $J: \mathbb{C} \rightarrow \mathbb{C}_T$.

Proof. Given an algebra $\varphi: T \Rightarrow \text{id}$ we define a functor $\overline{\varphi}: \mathbb{C}_T \rightarrow \mathbb{C}$ by $X \mapsto X$ and $f \mapsto \varphi \circ f$. This clearly yields a functor, and a retraction since:

$$\overline{\varphi}(J(f)) = \overline{\varphi}(\eta \circ f) = \varphi \circ \eta \circ f = f.$$

Conversely, given a retraction $K: \mathbb{C}_T \rightarrow \mathbb{C}$ of J , we define $\overline{K}: T \Rightarrow \text{id}$ via $\overline{K}_X = K(\text{id}_{TX}: TX \rightarrow X): TX \rightarrow X$. This yields a natural transformation, and a monad-algebra:

$$\begin{aligned} \overline{K}_X \circ T(\overline{K}_X) &= \overline{K}_X \circ \overline{K}_{TX} \\ &= K(\text{id}) \circ K(\text{id}) \\ &= K(\mu \circ T(\text{id}) \circ \text{id}) \\ &= K(\mu) \\ \overline{K}_X \circ \eta_X &= K(\text{id}) \circ KJ(\eta) \\ &= K(\mu \circ T(\text{id}) \circ \eta \circ \eta) \\ &= K(\eta) \\ &= \text{id} \end{aligned} \quad \begin{aligned} &= \overline{K}_X \circ \overline{K}_{TX} \\ &= K(\text{id}) \circ K(\text{id}) \\ &= K(\mu \circ T(\text{id}) \circ \text{id}) \\ &= K(\mu) \\ &= K(\mu \circ T(\text{id}) \circ \eta \circ \mu) \\ &= K(\text{id}) \circ KJ(\mu) \\ &= \overline{K}_X \circ \mu. \end{aligned} \quad \square$$

4. FREYD CATEGORIES ARE KLEISLI CATEGORIES

For monads and comonads, the Kleisli construction is characterised 2-categorically as a certain left 2-adjoint [11]. In this section we show that the (bijective) mapping $A \mapsto (\mathbb{C} \rightarrow \mathbb{C}_A)$ in the last section—from an Arrow to a Freyd category—allows for a similar 2-categorical characterisation. Therefore the bijective mapping is justifiably called the Kleisli construction for Arrows. In its course the notion of Arrows is extended to be defined on Freyd categories in general, instead of on categories with finite products. We will also make some additional accounts on the 2-categorical properties of this extended notion of Arrows.

Let us recall the situation for monads. The Kleisli construction is described in the following 2-adjunction, as the left 2-adjoint of the canonical “insertion” 2-functor Ins .

$$\text{Cat} \begin{array}{c} \xrightarrow{Ins} \\ \top \\ \xleftarrow{\mathcal{Kl}} \end{array} \text{Mnd}(\text{Cat}_*) \quad (11)$$

Here the 2-category $\text{Mnd}(\text{Cat}_*)$ is such that:²

- An object is a pair (\mathbb{C}, T) of a category \mathbb{C} and a monad T on it;
- A 1-cell $(H, \sigma): (\mathbb{C}, T) \rightarrow (\mathbb{D}, S)$ is a pair of a functor $H: \mathbb{C} \rightarrow \mathbb{D}$ and a natural transformation $\sigma: HT \Rightarrow SH$, which is compatible with monad structures of T and S .
- A 2-cell $\alpha: (H, \sigma) \Rightarrow (H', \sigma')$, is a natural transformation $\alpha: H \Rightarrow H'$ which is compatible with σ and σ' in a suitable sense.

The functor Ins is a canonical one mapping an object \mathbb{C} to (\mathbb{C}, id) . The functor \mathcal{Kl} of the Kleisli construction maps an object (\mathbb{C}, T) to the Kleisli category \mathbb{C}_T .

We shall do the same 2-categorical business for Arrows. Let us denote by FPCat the 2-category of categories with finite products. We may try to get a functor $\mathcal{Kl}: \text{Arr}(\text{FPCat}) \rightarrow \text{FPCat}$ for

²The notation $\text{Mnd}(\text{Cat}_*)$ comes from [11]. The 2-category Cat_* is obtained from Cat by reversing 2-cells. The constructor Mnd is actually a 2-functor which maps a 2-category \mathbb{C} to the 2-category of monads on \mathbb{C} .

a suitably defined $\mathbf{Arr}(\mathbf{FPCat})$, which maps (\mathbb{C}, A) to \mathbb{C}_A . However, the category \mathbb{C}_A does not necessarily have finite products: it has only a weaker structure of a Freyd category. This is the same difficulty which occurs already when we try to get the 2-adjunction (11) for strong monads.

This problem is mended by considering Arrows on Freyd categories. Hence the 2-adjunction (11) for Arrows would look as follows.

$$\mathbf{Freyd} \begin{array}{c} \xrightarrow{Ins} \\ \top \\ \xleftarrow{Kl} \end{array} \mathbf{Arr}(\mathbf{Freyd}) \quad (12)$$

The definition of Arrows on Freyd categories is just the same as on categories with finite products, except for the conditions on \mathbf{first} . Recall that in a Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{K}$, \mathbb{K} has a pre-monoidal structure which we denote by \boxtimes .

Definition 9 (Arrows on Freyd categories) An **Arrow** on a Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{K}$ is a monoid $A \otimes A \rightrightarrows A \xleftarrow{\mathbf{arr}} \mathbf{Hom}$ in the monoidal category $[\mathbb{K}^{\text{op}} \times \mathbb{K}, \mathbf{Sets}]$ of bifunctors, equipped with: for each $X, Y, Z \in \mathbb{K}$,

$$\mathbf{first}_{X,Y,Z} : A(X, Y) \rightarrow A(X \boxtimes Z, Y \boxtimes Z) ,$$

which is natural in X, Y and dinatural in Z . The operations should satisfy the following equations.

$$\mathbf{first}(a) \ggg \mathbf{arr}(J(\pi_1)) = \mathbf{arr}(J(\pi_1)) \ggg a \quad (4')$$

$$\mathbf{first}(a) \ggg \mathbf{arr}(\text{id} \boxtimes J(f)) = \mathbf{arr}(\text{id} \boxtimes J(f)) \ggg \mathbf{first}(a) \quad (5')$$

$$\mathbf{first}(a) \ggg \mathbf{arr}(\alpha) = \mathbf{arr}(\alpha) \ggg \mathbf{first}(\mathbf{first}(a)) \quad (6')$$

$$\mathbf{first}(\mathbf{arr}(g)) = \mathbf{arr}(g \boxtimes \text{id}) \quad (7')$$

$$\mathbf{first}(a \ggg b) = \mathbf{first}(a) \ggg \mathbf{first}(b) \quad (8')$$

The conditions (4')–(8') correspond to (4)–(8) in the original definition. The conditions (1)–(3) are gone because they amount to saying that A is a monoid. The associativity isomorphisms α in (6') are inherited from \mathbb{C} as $\alpha^{\mathbb{K}} = J(\alpha^{\mathbb{C}})$; this is because J preserves pre-monoidal structures. Recalling the intuition that a morphism in \mathbb{C} is a pure function while one in \mathbb{K} is an effectful one, (5') requires only pure functions commute with $\mathbf{first}(a)$.

For Arrows on categories with finite products, the operation \mathbf{first} is equivalently described as an “internal strength” $\text{ist} : A(X, Y) \rightarrow A(X, Y \times X)$ subject to suitable coherence conditions. This is also the case for Arrows on Freyd categories: however in the sequel we prefer to use \mathbf{first} .

Let us continue by describing ingredients of the adjunction (12).

The 2-category \mathbf{Freyd} of Freyd categories is defined in the following obvious way.

- An object is a Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{K}$.
- A 1-cell $(F, H) : (\mathbb{C} \xrightarrow{J} \mathbb{K}) \rightarrow (\mathbb{D} \xrightarrow{I} \mathbb{L})$ is a pair of a functor $F : \mathbb{C} \rightarrow \mathbb{D}$ preserving finite products and a functor $H : \mathbb{K} \rightarrow \mathbb{L}$ preserving pre-monoidal structures, such that $IF = HJ$.
- A 2-cell $(\alpha, \beta) : (F, H) \Rightarrow (F', H') : (\mathbb{C} \xrightarrow{J} \mathbb{K}) \rightarrow (\mathbb{D} \xrightarrow{I} \mathbb{L})$ is a pair of natural transformations $\alpha : F \Rightarrow F'$ and $\beta : H \Rightarrow H'$ such that $I\alpha = \beta J$.

The 2-category $\mathbf{Arr}(\mathbf{Freyd})$ of Arrows on Freyd categories is as follows.

- An object is a pair $(\mathbb{C} \xrightarrow{J} \mathbb{K}, A)$ of a Freyd category an Arrow A on that.

- A 1-cell $(F, H, \sigma) : (\mathbb{C} \xrightarrow{J} \mathbb{K}, A) \rightarrow (\mathbb{D} \xrightarrow{I} \mathbb{L}, B)$ is a 1-cell $(F, H) : (\mathbb{C} \xrightarrow{J} \mathbb{K}) \rightarrow (\mathbb{D} \xrightarrow{I} \mathbb{L})$ of Freyd together with a natural transformation

$$\begin{array}{ccc} \mathbb{K}^{\text{op}} \times \mathbb{K} & \xrightarrow{H^{\text{op}} \times H} & \mathbb{L}^{\text{op}} \times \mathbb{L} \\ & \searrow \scriptstyle A \quad \xRightarrow{\sigma} \quad \swarrow \scriptstyle B & \\ & \text{Sets} & \end{array},$$

which is compatible with arr , \ggg and first , in the following sense.

- $\sigma \bullet \text{arr}^A = (\text{arr}^B \circ (H^{\text{op}} \times H)) \bullet \text{Hom}(H)$. Here $\text{Hom}(H)$ is the following natural transformation induced by H 's action on morphisms.

$$\begin{array}{ccc} \mathbb{C}^{\text{op}} \times \mathbb{C} & \xrightarrow{H^{\text{op}} \times H} & \mathbb{D}^{\text{op}} \times \mathbb{D} \\ & \searrow \scriptstyle \text{Hom}_{\mathbb{C}} \quad \xRightarrow{\text{Hom}(H)} \quad \swarrow \scriptstyle \text{Hom}_{\mathbb{D}} & \\ & \text{Sets} & \end{array}$$

- $\sigma \bullet \ggg^A = (\ggg^B \circ (H^{\text{op}} \times H)) \bullet (\sigma \otimes \sigma)$, where the natural transformation $\sigma \otimes \sigma$ is defined in the following obvious manner.

$$\begin{array}{ccc} (A \otimes A)(X, Y) & \xrightarrow{(\sigma \otimes \sigma)_{X, Y}} & (B \otimes B)(HX, HY) \\ \uparrow & & \uparrow \\ \coprod_{P \in \mathbb{C}} A(X, P) \times A(P, Y) & \xrightarrow{\quad} & \coprod_{Q \in \mathbb{D}} B(HX, Q) \times B(Q, HY) \\ \uparrow \scriptstyle \kappa_P & & \uparrow \scriptstyle \kappa_{HP} \\ A(X, P) \times A(P, Y) & \xrightarrow{\sigma_{X, P} \times \sigma_{P, Y}} & B(HX, HP) \times B(HP, HY) \end{array}$$

Note that this construction of $\sigma \otimes \sigma : A \otimes A \Rightarrow (B \otimes B) \circ (H^{\text{op}} \times H)$ is not an instance of the functoriality of \otimes .

- With first : for each X, Y, Z in \mathbb{C} ,

$$\begin{array}{ccc} A(X, Y) & \xrightarrow{\sigma} & B(HX, HY) \\ \text{first}^A \downarrow & & \downarrow \text{first}^B \\ A(X \boxtimes Z, Y \boxtimes Z) & \xrightarrow{\sigma} B(H(X \boxtimes Z), H(Y \boxtimes Z)) \xrightarrow{\cong} B(HX \boxtimes HZ, HY \boxtimes HZ) \end{array}$$

where the isomorphism is because H preserves pre-monoidal structures.

- A 2-cell $(\alpha, \beta) : (F, H, \sigma) \Rightarrow (F', H', \sigma') : (\mathbb{C} \xrightarrow{J} \mathbb{K}, A) \rightarrow (\mathbb{D} \xrightarrow{I} \mathbb{L}, B)$ is a 2-cell $(\alpha, \beta) : (F, H) \Rightarrow (F', H')$ of Freyd which is compatible with σ and σ' in the following sense.

$$\begin{array}{ccc} \mathbb{K}^{\text{op}} \times \mathbb{K} & \xrightarrow{H^{\text{op}} \times H'} & \mathbb{L}^{\text{op}} \times \mathbb{L} \\ & \searrow \scriptstyle A \quad \xRightarrow{\sigma} \quad \swarrow \scriptstyle B & \\ & \text{Sets} & \end{array} = \begin{array}{ccc} \mathbb{K}^{\text{op}} \times \mathbb{K} & \xrightarrow{H^{\text{op}} \times H'} & \mathbb{L}^{\text{op}} \times \mathbb{L} \\ & \searrow \scriptstyle A \quad \xRightarrow{\sigma'} \quad \swarrow \scriptstyle B & \\ & \text{Sets} & \end{array} \quad (13)$$

Here note that a natural transformation $\beta : H \Rightarrow H' : \mathbb{K} \rightarrow \mathbb{L}$ induces its dual $\beta^{\text{op}} : H'^{\text{op}} \Rightarrow H^{\text{op}} : \mathbb{K}^{\text{op}} \rightarrow \mathbb{L}^{\text{op}}$, with the direction of \Rightarrow reversed.

The 2-functor $\text{Ins} : \text{Freyd} \rightarrow \text{Arr}(\text{Freyd})$ acts as follows.

- An object $\mathbb{C} \xrightarrow{J} \mathbb{K}$ is mapped to $(\mathbb{C} \xrightarrow{J} \mathbb{K}, \text{Hom}_{\mathbb{K}})$. The bifunctor $\text{Hom}_{\mathbb{K}}$ is obviously an Arrow: its operation first comes from the pre-monoidal structure of \mathbb{K} .
- A 1-cell (F, H) is mapped to $(F, H, \text{Hom}(H))$. Compatibility of $\text{Hom}(H)$ with first is because H preserves pre-monoidal structures.

- A 2-cell $(\alpha, \beta) : (F, H) \Rightarrow (F', H')$ is mapped as it is. The compatibility of β with $\text{Hom}(H)$ and $\text{Hom}(H')$ amounts to the naturality of $\beta : H \Rightarrow H'$.

The 2-functor $\mathcal{K}\ell : \mathbf{Arr}(\mathbf{Freyd}) \rightarrow \mathbf{Freyd}$ is essentially the Kleisli construction for Arrows in Definition 1. Namely,

- An object $(\mathbb{C} \xrightarrow{J} \mathbb{K}, A)$ is mapped to a Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{K} \xrightarrow{J_A} \mathbb{K}_A$, where $\mathbb{K} \xrightarrow{J_A} \mathbb{K}_A$ is constructed like in Definition 1. The composition $J_A \circ J$ is indeed a Freyd category: for example it preserves central morphisms essentially due to the Arrow law (5').
- A 1-cell $(F, H, \sigma) : (\mathbb{C} \xrightarrow{J} \mathbb{K}, A) \rightarrow (\mathbb{D} \xrightarrow{J} \mathbb{L}, B)$ induces a functor $\tilde{H} : \mathbb{K}_A \rightarrow \mathbb{L}_B$ such that: $\tilde{H}X = HX$ on objects and $\tilde{H}f = \sigma_{X,Y}(f)$ on morphisms.
- A 2-cell (α, β) is mapped to $(\alpha, \tilde{\beta})$, where a component $\tilde{\beta}_X$ is given by $\text{arr}(\beta_X)$. The naturality of $\tilde{\beta}$ amounts to the coherence condition (13).

Theorem 10 *We have a 2-adjunction $\mathcal{K}\ell \dashv \text{Ins} : \mathbf{Arr}(\mathbf{Freyd}) \rightarrow \mathbf{Freyd}$ as in Diagram (12).*

Proof. Its unit is given by $(\mathbb{C} \xrightarrow{J} \mathbb{K}, A) \xrightarrow{(\text{id}, J_A, \iota)} (\mathbb{C} \xrightarrow{J} \mathbb{K} \xrightarrow{J_A} \mathbb{K}_A, \text{Hom}_{\mathbb{K}_A})$, where ι is a canonical natural transformation whose component is given by $\text{id} : A(X, Y) \rightarrow \text{Hom}_{\mathbb{K}_A}(X, Y)$. \square

In relation to Arrows on categories with finite products—which form a 2-category $\mathbf{Arr}(\mathbf{FPCat})$ in a way similar to $\mathbf{Arr}(\mathbf{Freyd})$ —we have the following situation with the obvious horizontal insertion 2-functors.

$$\begin{array}{ccc} \mathbf{Arr}(\mathbf{Freyd}) & \xleftarrow{\text{Ins}'} & \mathbf{Arr}(\mathbf{FPCat}) \\ \mathcal{K}\ell \downarrow \uparrow \text{Ins} & & \uparrow \text{Ins} \\ \mathbf{Freyd} & \xleftarrow{\text{Ins}} & \mathbf{FPCat} \end{array} \quad (14)$$

In the rest of this section we elaborate further on this diagram. First, the correspondence result from [3] is incorporated.

Theorem 11 *The bijective correspondence in Theorem 2—between Freyd categories and Arrows on categories with finite products—extends to an isomorphism between 2-categories \mathbf{Freyd} and $\mathbf{Arr}(\mathbf{FPCat})$ in an obvious way.*

$$\mathbf{Freyd} \begin{array}{c} \xrightarrow{\text{Arr}} \\ \xleftarrow[\cong]{\mathcal{K}\ell} \end{array} \mathbf{Arr}(\mathbf{FPCat}) \quad \square$$

In Diagram (11) for Monads, the insertion functor Ins has a right 2-adjoint given by the obvious forgetful functor. This is also the case for Arrows.

Proposition 12 *Each of the “insertion” 2-functors in Diagram (14) has a right 2-adjoint. For those named Ins it is given by the obvious forgetful functor; the right 2-adjoint of Ins' follows from the isomorphism in Theorem 11.* \square

An intuition on an object $(\mathbb{C} \xrightarrow{J} \mathbb{K}, A)$ of $\mathbf{Arr}(\mathbf{Freyd})$ is that it has two different levels of extra computational structures added to \mathbb{C} . One is described by the Freyd category $\mathbb{C} \xrightarrow{J} \mathbb{K}$, and on top of it we have the other one expressed as the Arrow A . By applying $\mathcal{K}\ell : \mathbf{Arr}(\mathbf{Freyd}) \rightarrow \mathbf{Freyd}$ we have these two structures combined and expressed as a composed Freyd category. In this sense, the 2-category $\mathbf{Arr}(\mathbf{Freyd})$ has extra expressive power—which is essentially redundant—compared to \mathbf{Freyd} . This point is put in precise, 2-categorical terms in the following two propositions.

Proposition 13 *\mathbf{Freyd} is a full reflective 2-subcategory of $\mathbf{Arr}(\mathbf{Freyd})$, with Ins its embedding.*

Proof. The 2-functor $Ins : \mathbf{Freyd} \rightarrow \mathbf{Arr}(\mathbf{Freyd})$ is full, faithful and injective on objects. In particular, the fullness follows from the compatibility of 1-cells of $\mathbf{Arr}(\mathbf{Freyd})$ with arr . \square

Aside from the insertion functor Ins , there is another canonical 2-functor $\mathbf{Freyd} \rightarrow \mathbf{Arr}(\mathbf{Freyd})$ which we shall call Arr . On objects,

$$\begin{array}{lll} Ins : & \mathbb{C} \xrightarrow{J} \mathbb{K} & \mapsto (\mathbb{C} \xrightarrow{J} \mathbb{K}, \text{Hom}_{\mathbb{K}}) , \\ Arr : & \mathbb{C} \xrightarrow{J} \mathbb{K} & \mapsto (\mathbb{C} \xrightarrow{\text{id}} \mathbb{C}, \text{Hom}_{\mathbb{K}}(J-, J+)) . \end{array}$$

Intuitively, Ins encodes the extra structure carried by J into a Freyd category; Arr does it into an Arrow.

Proposition 14 *We have the following 2-adjunctions.*

$$\begin{array}{ccc} & Ins & \\ & \downarrow & \\ \mathbf{Freyd} & \xleftarrow{\mathcal{K}\ell} & \mathbf{Arr}(\mathbf{Freyd}) \\ & \uparrow & \\ & Arr & \end{array}$$

Proof. Each object $(\mathbb{C} \xrightarrow{J} \mathbb{K}, A)$ of $\mathbf{Arr}(\mathbf{Freyd})$ induces the following two universal morphisms. Here the canonical natural transformations ι and ι' have the identity maps as their components.

$$(\mathbb{C} \xrightarrow{\text{id}} \mathbb{C}, A(J-, J+)) \xrightarrow{(\text{id}, J, \iota')} (\mathbb{C} \xrightarrow{J} \mathbb{K}, A) \xrightarrow{(\text{id}, J_A, \iota)} (\mathbb{C} \xrightarrow{J} \mathbb{K} \xrightarrow{J_A} \mathbb{K}_A, \text{Hom}_{\mathbb{K}_A})$$

They form the unit/counit of the 2-adjunctions. \square

Remark 15 *We may aim at a similar 2-categorical characterisation of Eilenberg-Moore algebras for Arrows. Here we shall show an attempt we have taken and how it fails.*

The idea comes from the duality between the Eilenberg-Moore and Kleisli constructions for monads. The Eilenberg-Moore construction for monads is characterised as the following right 2-adjoint [11].

$$\begin{array}{ccc} & Ins & \\ & \downarrow & \\ \mathbf{Cat} & \xleftarrow{\perp} & \mathbf{Mnd}(\mathbf{Cat}) \\ & EMAlg & \end{array}$$

This is a dual of the 2-adjunction (11) for Kleisli in the following sense. A 1-cell $(H, \sigma) : (\mathbb{C}, T) \rightarrow (\mathbb{D}, S)$ of $\mathbf{Mnd}(\mathbf{Cat})$ has its component $\sigma : SH \Rightarrow HT$, in the reversed direction from $\mathbf{Mnd}(\mathbf{Cat}_)$. We may try the same dualisation of the adjunction (12).*

The problem lies in defining the dualised version of Ins . It should map a 1-cell (F, H) in \mathbf{Freyd} to (F, H, σ) , with some canonical natural transformation σ such that

$$\begin{array}{ccc} \mathbb{C}^{\text{op}} \times \mathbb{C} & \xrightarrow{H^{\text{op}} \times H} & \mathbb{D}^{\text{op}} \times \mathbb{D} \\ & \xleftarrow{\sigma} & \\ \text{Hom}_{\mathbb{C}} & \xrightarrow{\quad} & \text{Sets} \quad \xleftarrow{\quad} \text{Hom}_{\mathbb{D}} \end{array}$$

It is not clear how we can get σ in this direction.

5. CONCLUSIONS AND FUTURE WORK

Starting from the observation of Arrows-as-monoids—extending monads-as-monoids—we have made a first step toward the notions of Kleisli categories and Eilenberg-Moore algebras for Arrows. The definitions have been presented with reasonable justifications.

In particular, the Kleisli category for an Arrow turns out to be the Freyd category that corresponds to the given Arrow one-to-one. This situation is put clearly in 2-categorical terms, accompanied with some additional observations.

The definition of Arrow-algebras in this paper we do not consider fully satisfactory. We might expect an algebra *carried by* specific object(s). Concrete examples of what should be called Arrow-algebras are to be found: those examples would also help us to get a more suitable definition.

In [8] Freyd categories are shown to be equivalent to the fibrational notion of κ -categories. The relation between κ -categories and the current work is yet to be elaborated.

ACKNOWLEDGEMENTS

Thanks are due to Chris Heunen, John Power, Tarmo Uustalu, Hiroshi Watanabe and the anonymous referees for helpful discussions and comments.

REFERENCES

- [1] A. Alimarine, S. Smetsers, A. van Weelden, M. van Eekelen, and R. Plasmeijer. There and back again: arrows for invertible programming. In *Proceedings of the 2005 ACM SIGPLAN workshop on Haskell*, pages 86–97. ACM Press, 2006.
- [2] F. Borceux. *Handbook of Categorical Algebra*, volume 50, 51 and 52 of *Encyclopedia of Mathematics*. Cambridge Univ. Press, 1994.
- [3] C. Heunen and B. Jacobs. Arrows, like monads, are monoids. In *Mathematical Foundations of Program Semantics*, Elect. Notes in Theor. Comp. Sci. 158, Elsevier, Amsterdam, 2006.
- [4] J. Hughes. Generalising monads to arrows. *Science of Comput. Progr.*, 37:67–111, 2000.
- [5] P. Jansson and J. Jeuring. Polytypic data conversion programs. *Science of Comput. Progr.*, 43:35–75, 2002.
- [6] S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 1971.
- [7] R. Paterson. A new notation for arrows. In *International Conference on Functional Programming (ICFP)*, volume 36(10), pages 229–240. ACM SIGPLAN Notices, 2001.
- [8] J. Power and H. Thielecke. Environments, continuation semantics and indexed categories. In M. Abadi and T. Ito, editors, *Theoretical Aspects of Computer Software*, number 1281 in Lect. Notes Comp. Sci., pages 391–414. Springer, Berlin, 1997.
- [9] J. Power and H. Watanabe. Combining a monad and a comonad. *Theor. Comp. Sci.*, 280(1–2):137–162, 2002.
- [10] E.P. Robinson and J. Power. Premonoidal categories and notions of computation. *Math. Struct. in Comp. Sci.*, 7(5):453–468, 1997.
- [11] R. Street. The formal theory of monads. *Journ. of Pure & Appl. Algebra*, 2:149–169, 1972.
- [12] T. Uustalu and V. Vene. The essence of dataflow programming. In *3rd Asian Symp. on Programming Languages and Systems, APLAS'05*, number 3780 in Lect. Notes Comp. Sci., pages 2–18. Springer, 2005.
- [13] T. Uustalu and V. Vene. Signals and comonads. *Journ. of Universal Comp. Sci.*, 11(7):1310–1326, 2005.